Islamic University of Technology (IUT)

Department of Computer Science and Engineering (CSE)

Image Captioning Using Scene Graph and Language Decoder

Authors

Asaduzzaman Herok, 170041034

Kawsar Ahmed, 170041021

Safayet Hossain Masum, 170041050

Supervisor

Dr. Md. Hasanul Kabir

Professor, Department of CSE

Co-Supervisor

Sabbir Ahmed

Lecturer, Department of CSE

A thesis submitted to the Department of CSE

in partial fulfillment of the requirements for the degree of B.Sc. Engineering in CSE

Academic Year: 2020-2021

April, 2022

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Asaduzzaman Herok, Kawsar Ahmed and Safayet Hossain Masum under the supervision of Dr. Md. Hasanul Kabir, Professor of Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Gazipur, Dhaka, Bangladesh and Sabbir Ahmed, Lecturer of Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Gazipur, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Authors:

	Kawsar Ahmed	Safayet
Asaduzzaman Herok	Kawsar Ahmed	Safayet Hossain Masum
Student ID: 170041034	Student ID: 170041021	Student ID: 170041050

Supervisor:

17.5.22

Dr. Md. Hasanul Kabir
Professor
Department of Computer Science and Engineering
Islamic University of Technology

Co-Supervisor:

17.5.22.

Sabbir Ahmed
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology

Acknowledgement

We would like express our gratitude towards IUT authority for providing assistance required to implement our proposed system. We are indebted to our professor, Dr. Md. Hasanul Kabir for providing us with insightful knowledge and guiding us at every stage of our journey. We are also indebted to our lecturer, Sabbir Ahmed for pushing us through every difficult situations. Finally, we would like to express our heartiest appreciation towards our family members for their continuous support, motivation, suggestions and help, without which we could not have achieved the scale of implementation that we have achieved.

Abstract

Image captioning refers to the task of assigning natural language description to an image from its visual and cognitive information. It's a multi-modal task where image understanding and natural language generation is the backbone. Real life applications like content based image retrieval, navigation of self driving car, assisting visually impaired people, visual question answering etc. are the areas where image captioning can be used. Even though a significant amount of research work has been done on image captioning, still a lot of works can be done to improve the accuracy of Image captioning systems specially for visually challenged images. We explored the possibilities of developing a more robust and accurate image captioning system that can handle motion blur, plain text tokens, partially visible objects in an image. We proposed a pipeline that includes Global feature extraction for extracting overall pictorial information of the image, Scene Graph for detecting objects and learning individual relationship among the objects, OCR token extractor for understanding the plain text in the image (if available) and an encoder-decoder based language model for features to text translation. The main goal was to exploit the research opportunities and improve the research gap. Finally, we explored the result of our findings and did a comparative analysis of our architecture with existing state-of-the-art papers on VizWiz-Captions dataset since images of this dataset are taken by visually impaired people making images more visually challenged.

Keywords— Image Captioning, Scene Graph, Encoder, Decoder, OCR

Contents

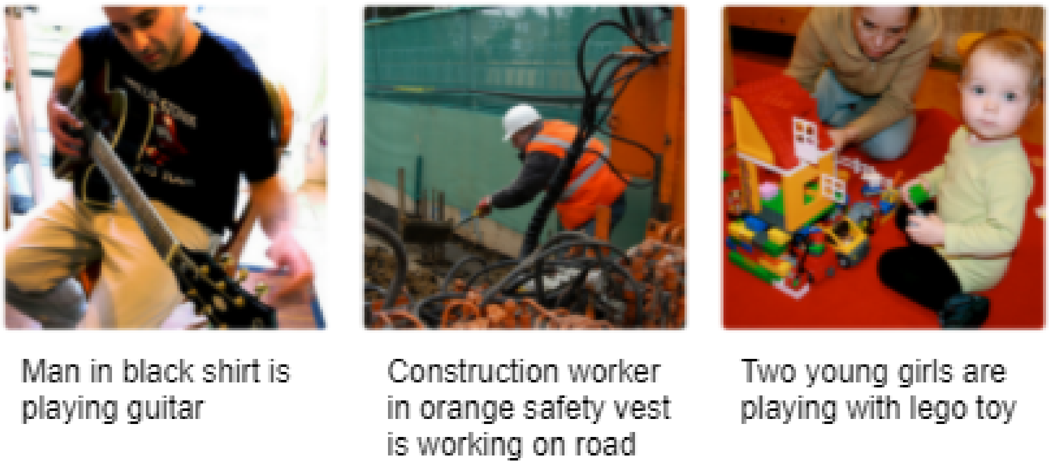
List of Tables

List of Figures

Chapter 1

Introduction

An image can speak a thousand words. Every day a large number of images are uploaded in the internet from different sources. But most of the sources do not include any explanation. Viewers need to interpret the images themselves. So, automated translation of an image to plain natural language is one of the most challenging tasks in the era of machine learning. This translation is known as Image captioning. To define formally, it is the task of describing the visual and cognitive information of an image in natural language (Figure: 1.1).



Man in black shirt is playing guitar

Construction worker in orange safety vest is working on road

Two young girls are playing with lego toy

Figure 1.1: Image Captioning Examples Taken from Microsoft COCO Captions [11]

There are plenty of application areas of image captioning. For example, automatic image indexing for content-based image retrieval, digital libraries, web searching. In social media, automated caption generation from image is also crucial. It is also popular in Artificial Intelligence research area where machines deal with image interpretation and human-level language understanding. Image captioning can also be used for visual question answering, where any natural language questions about the image are answered through an automated system. Generating medical report in natural language from medical images can also be done using image captioning.

In the world, around 253 million of people are visually impaired in total [15]. They cannot understand image contents without help of others, let alone perceive their surroundings. An automated captioning system can give them the ability to perceive by hearing the captions through a text to speech medium. This idea is opening a new door to see the world for the visually impaired people.

A good number of research works have been worked out in the image captioning area [16].From deep learning-based approaches with Recurrent Neural Networks (RNNs) fed with global image features, algorithms are evolving with attentive approaches and reinforcement learning. With the breakthroughs of Transformers and self-attention in recent years, working pipelines consist of object detection and classification, individual object features, global image features, object to object relationship extraction, attention mechanism, language decoders like GPT [17]. To evaluate how much the captions are aligned with ground-truth captions, Natural Language Processing communities have defined some evaluation metrics such as BLEU, METEOR, etc.

Using crowd-sourcing a good number of image captioning datasets are publicly available, such as Flickr30k [18], MS-COCO Captions [11], Visual Genome. Most of them consist of images taken by real users, not by visually impaired people. As a result, there's a tendency of producing generic captions. For this reason, existing image captioning algorithms' applicability is restricted. They can't be deployed widely in practical real-world settings which include assisting people who are visually impaired in navigating and accomplishing everyday tasks. Because of these limitations, goal-oriented image captioning challenges emerged: *VizWiz Challenge for captioning images taken by people who are blind* [5].

Since VizWiz challenge is very recent and people are still working on it, only few works have come to the light. Pierre Dognin et al. incorporated OCR token to include plain text of image in captions [6]. Lun Huang et al. proposed double attention mechanism with LSTM recurrent neural net to produce captions [9]. In extend of AoA-NET [9], Dong Wook Kim et al. introduced a graph convolution network and pre-trained language model to enhance performance [7]. Other works can be seen in the VizWiz challenge leader-board but corresponding research works are not found.

Though current works are doing great in automated image captioning, still some important improvements can be done on generated captions. Some problems with current models used for image captioning includes missing object relation in the generated captions, missing plain text tokens, missing partially visible objects. Current models implicitly learn object to object relationships but explicitly defined relationship suppose to give better context for natural language generation. Pre-extracted OCR tokens can be incorporated in the final captions. Some models have started to use this idea. It's obvious that just generating word sequence is not captioning. The captions should be grammatically and semantically correct. In this regard pre-trained language decoders works good for semantically and grammatically correct sentences.

To give language models object to object relationships explicitly Xu Yang et al. [19] used Scene Graph to represent object to object relationships. Realising the impact of scene graph on language decoder, we are proposing to use scene graph in our pipeline.

Our final pipeline consists of Scene Graph [14] for explicit object to object relations, inceptionV3 [20] for global feature extraction, Paddle OCR for OCR token extraction and encoder-decoder based language model i.e vanilla transformer [21]. In this work we will explore the opportunities of our pipeline and do a comparative analysis of our architecture with existing State-of-the-Art models.

1.1 Problem Statement

Image Captioning is the task of describing visual and cognitive information of that image in layman natural language. So the problem is to develop a robust and accurate image captioning architecture that can handle several challenges like motion blur, plain text tokens, partially visible objects in the image. The architecture will take an image and generate a caption which should be understandable by layman.

1.2 Challenges

The main challenge of image captioning is the interpretation of the image by the machine. Both the spatial and semantic information are crucial for understanding the underline cognitive and visual relations. How the image is captured, surrounding environment have great impact on the image visibility and quality. Based on these we can divide the challenges into two categories: fine grained image captioning and visually challenged image captioning.

1.2.1 Fine Grained Image Captioning

Clear, vivid images are easy for machine to interpret. The bench-marked datasets MSCOCO [11], Flickr [18], Visual Genome [22] are well known datasets of clear, well focused images. Still it has some challenges on both visual perspective and natural language perspective.

- Linking background contents with the foreground contents.

- Identifying novel objects.
- Extracting plain text, interpreting traffic signs, understanding bill-board directions/advertisement.
- Keeping naturalness in caption generation.
- Incorporating plain text, signs of images into the caption
- Generalization of the captioning system.

1.2.2 Visually Challenged Image Captioning

Not every human in the world is a good photographer. Beside environmental condition differs too much. Even images are captured by visually impaired people. Sometimes human themselves cannot understand the visual content of the image. Some examples are shown in figure 1.2

- Focus-less images are hard to interpret.
- Motion blurred image are very common.
- Lack of luminance or over brightness of images
- Partially visible objects.
- Imbalanced shadow effects.
- Indoor images often have low lighting condition, novel objects and lots of plain texts which may be very small in font size.
- Linking novel objects with known objects.

Because of these challenges object detectors suffer very much. But correct object detection is the heart of accurate captions. Beside each relations among the objects defines the cognitive understanding for the captions. Consisting of these kind of visually challenged images and crowd-sourced captions, a dataset of size 39000 is publicly available. All the images of this dataset is captured by visually impaired people. This makes automated image captioning more hard. Though a lots of work have been done in the field of fine grained image captioning, robust and accurate visually challenged captioning is lagging behind.

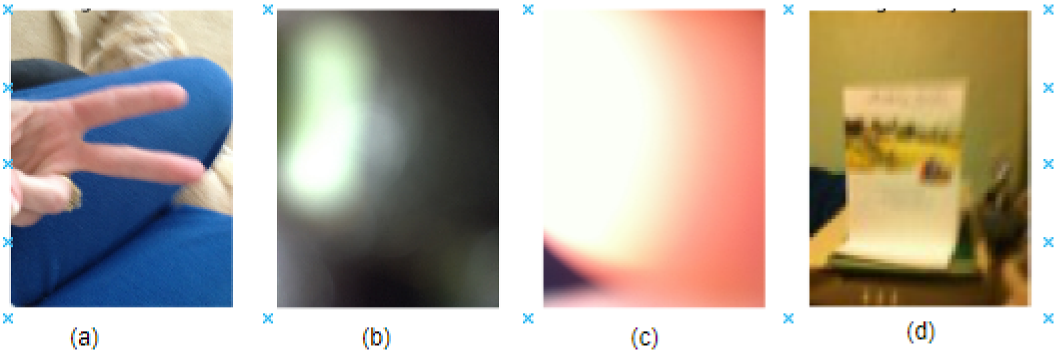


Figure 1.2: Images captured by Blind people taken from VizWiz-Captions [5]

1.3 Objectives

Our main objective is to develop more robust and accurate captioning system for visually challenged images. Since a lot of works have been done for fine grained images, our main focus is on visually challenged images. To address this issue, our solution should handle partially visible objects, improve object to object relationships, if there is plain text in the image those should be reflected on the generated captions. If these objectives are full filled then we would focus on building lighter models for small scale devices.

1.4 Contribution

Our main contribution was our proposed pipe-line. Use of the scene graphs, OCR tokens, global features are not novel individually. But our pipelines creates a fusion of them to improve on the research opportunities found which hasn't been explored before. Besides, use of explicit scene graph generator on VizWiz-caption is also new. We found some research opportunities, proposed a pipeline that we believe should overcome them and explore the opportunities along with creating new opportunities to explore.

1.5 Organization of Thesis

This dissertation shows the evolution of automated image captioning systems, some research opportunities and finally proposal and analysis of a pipeline that might overcome those research gaps.

In Chapter 2 titled as Background study we discuss some background study we have done related to our works. We also discuss some terms and methods related to our work. And finally we discuss some of the most related publications their architecture and contribution.

In Chapter 3 titled as System Architecture, we discuss our proposed pipeline and how it's supposed to overcome the research gaps found. It contains our architecture, description of each module and how each of the module works and are related to each other.

In Chapter 4 titled as Results Analysis and Discussion, we present the results of our experiments in different setups. We present our result in popular evaluation matrices for easier understanding. We also do a analysis of our result mainly focusing on findings from the result. Finally we compare our result with some state of the art results in this field.

In Chapter 5 titled as Conclusion and Future Work, we discuss our findings and contribution from the experiments and research we have done. We also discuss some research opportunities for future.

Chapter 2

Background Study

Before working on the image captioning system,we had done some background study of core elements of captioning like Word Embedding, Seq2Seq, Attention Mechanism, RNN, Long-Short Term Memory (LSTM), Transformer. The following sections will give an overview of each of these elements. Later sections well show us some of the most exciting works done so far on image captioning.

2.1 Word Embedding

Logical numeric representations of texts are called word embedding. The machine must learn human language, which is one of the initial problems in natural language processing. This is crucial because as humans, we are capable of perceiving and comprehending languages. We know the difference between ‘king’ and ‘queen’ as well as ‘man’ and ‘woman’ In the case of computers, though, it is less evident. The idea is to create a high-dimensional vector space in which each word has its own unique vector representation [23].

A survey [23] on vector space modeling for words meaning shows how vectors represents words for machines. Earlier works used one hot vector representation for words. Each word was then represented with a large vector with a single 1 in one position and others are zero. It didn't help much. Later Word2Vec [24], Glove word [25] embedding evolved. Words with similar meaning have similar representations or being closer in the vector space. The vector dimension reduced and each entry of the a particular value represents a feature of the particular word [26].

There are a few more important factors to think about. Words have multiple meanings. Some words may have distinct meanings in different circumstances. In this circumstance, modern word embedding has distinct representations for the same words with different meanings. For example, ‘I always park my automobile in the park.’ Because each ‘park’ in this phrase has a distinct meaning, their embedding should differ. The word ‘parking’ in the first line of the phrase ‘Drive through the parking lot’ must be the most closely linked. The word ‘Park’ is more directly related with the park mentioned in the first phrase in the statement ‘Mr. Raj always longed to visit the Ramna Park’.

2.2 Seq2Seq

Google [27] proposed Sequence to Sequence modeling, or Seq2Seq, as an end-to-end sequence learning approach. It was largely used for text translation from one language to another using neural machine translation. It was then applied to text summarization, sentiment conversion, and other sequential models. Sentences are essentially a group of words that can be transformed into word vectors or embedded. In terms of architecture, Seq2Seq models have two components: the encoder and the decoder.

The encoders go through the input sequence in order. As encoders, RNNs [28] or LSTMs [29] are utilized. Each input vector is processed by an encoder RNN, which then provides a hidden representation to the next layer along with the next input vector. As a result, each RNN output has all of the information from the previous input sequences. As a result, a context vector (encoder representation) is created. After that, the model utilizes the Decoder to decode the context vector into the recommended output sequence, which is also a series of RNNs or LSTMs. The decoder RNNs decode the context vector in a sequential manner, producing one output vector at a time.

2.3 Attention Mechanism

The sequence data is computed one by one in the Encoder Decoder model. The starting information in lengthier phrases is more likely to be lost due to the context vector’s long correlated inter-dependencies. This is why the attention mechanism was created [30]. The encoder vector is created by converting the words ’How,’ ’Are,’ and ’You’ into word embedding.

Apart from the context vector, the decoder is additionally given a direct link from the input sequence representing the relevance of the corresponding words, indicating where the model should concentrate its efforts in order to accurately predict the output sentence. The invention of attention mechanism helped to improve the sequence models significantly [13], [31].

2.4 Recurrent Neural Network

Recurrent Neural Network(RNN) [28] is a class of neural networks that works on sequential data. RNN can be thought as simple feed forward network with some fixed memory unit. RNN is recurrent because it takes each data and runs the same function on it. The generated output is then feed to network again making each execution dependent on the previous computation. Since the output of previous time step is feed back to the function with new data, RNN has information about the previous data and their possible relationship. It helps the network understand sequence data. Conventional feed-forward netword doesn't have any memory unit. But RNNs have internal state memory, enabling it to handle sequence data. That's why RNNs [28] were the building blocks of sequence data processing, like handwriting recognition, speech recognition, machine translations, caption generations.

There are mainly four types of RNN [32]:

- One to One RNN
- One to Many RNN
- Many to One RNN
- Many to Many RNN

2.5 Long-Short Term Memory

LSTM [29] or Long-Short Term Memory was first introduced to overcome the exploding and vanishing gradient problems of RNN architecture. RNNs [28] tend to forget previous data as it runs for longer time. LSTMs are utilized to help with short-term memory recall. When new information is added to RNN, it fully alters the previous information. RNN is incapable of distinguishing between important and unimportant data. When new information is added to LSTM, there is only a minor change in current information because LSTM incorporates gates that control the flow of information. The gates determine whether data is significant and will be helpful in the future, as well as which data must be deleted.

The three gates used in LSTM architectures are:

- Input gate
- Output gate
- Forget gate

2.6 Transformer

The Transformer [21] model offered a new network architecture based entirely on the attention mechanism for sequence to sequence modeling. Sequential models like RNN [28] and LSTM [29] process sentences word by word, posing a significant issue in the parallelization process. They included multi-head attention into their design. It's essentially a self-attention implementation, in which the input sequence learns the similarities among itself, indicating which words are more linked to each other. As a result, it will have a better understanding of the language and how it is constructed. Three vector variables, Q (query), K (key), and V (value), are used to train multi-headed attention (value). The link between the word vectors is largely included in these parameters.One of the reasons sequential models like RNN have been so successful in language processing is that it goes over each word in the phrase one by one, learning the positional relationship between them. Positional Encoding was developed in Transformer to address this issue. For each odd and even position, a sin and cos function are used to maintain it. Transformers can theoretically save infinitely long connection relationships among word vectors given adequate GPU computational capacity, whereas RNN/LSTM longer sequences tend to lose their original information. Furthermore, because the data is not handled one by one, parallelization is particularly efficient.

2.7 Related Works on Image Captioning

Early image captioning systems were template based [12] and retrieval based [12]. But in this deep-learning era all works are done using deep-learning solutions. First part of the image captioning pipeline is providing effective features from visual information i.e visual encoding. We can categorize current deep-learning approaches of visual encoding in 5 broader categories. *1. Non-attentive Global CNN feature based, 2 Attention Over CNN Features Grid, 3. Attention Over Visual Regions, 4. Encoding based on Scene Graph, 5. Encoding with Self-Attention* . Overview shown in figure 2.1

Earlier deep-learning based approaches started with Global CNN feature based captioning. In this approach a pre-trained CNN is used to extract global features from images. These are the features which are used as conditioning elements for the language model. For example, *Oriol Vinyals et al.* [33] Show and tell] used pretrained GoogleNet [34] as feature extractor, *Karpathy et al.* [35] used global features extracted from AlexNet [36] as the input for a language decoder.

Next deep-learning practice for image captioning was Attention over CNN feature Grid followed by Attention Over Visual Regions. Human brain integrates top-down cognitive reasoning

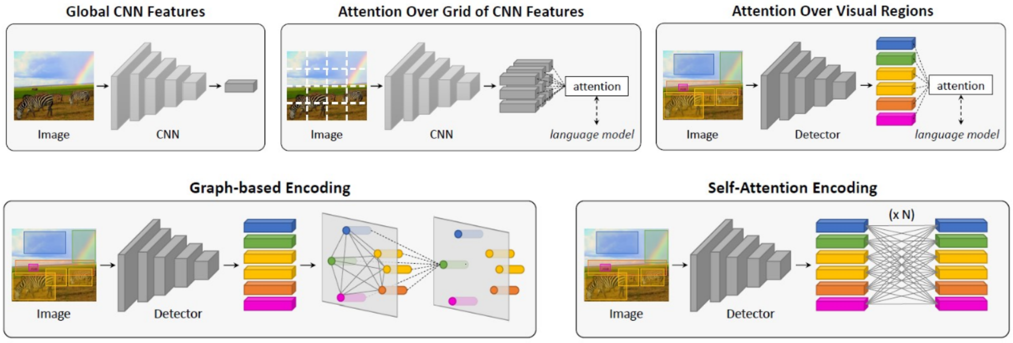


Figure 2.1: Deep Learning approaches of Image Captioning {Image Source: [12]}

and bottom-up constant flow of visual information to understand image content. Attention Over Visual Regions incorporate bottom-up visual information and language decoder acts as top-down reasoning. *Anderson et al.* [1] proposed first this bottom-up approach which is still used in many image captioning system as region feature extractor.

To learn relationships of objects in images, next deep-learning approach is Scene Graph parsing. It is also seen to be used in current image captioning systems. *Xu Yang et al. 2019* [19] proposed an auto encoding of scene graph for learning object to object relationship so that the language model doesn't need to learn implicitly. As a special case of graph-based encoding, *Yao et al.* employed a tree to represent the image as a hierarchical structure and then fed it to a Tree-LSTM. [37]

In 2017, *Vaswani et al.* [21], presented self attention for machine translation, language understanding, language generation. The full model then called the Transformer architecture in NLP domain. Later it was used on other domains like computer vision's Vision Transformer. *Huang et al.* [9] presented an extension of the attention operation, named "Attention on Attention".

LSTM-based decoders were once popular for language models. However, transformer-based decoders are currently the most used. Because image captioning task can be viewed as a sequence-to-sequence generation problem where image regions are feed as sequence. The Transformer networks are serving better in this regard.

2.7.1 Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering [1]

In this work, *Peter Anderson et al.* [1] presented a method that can calculate attention down to the objects and visual regions of images. Previously, image captioning for image understanding relied solely on top-down visual attention techniques. The bottom-up approach presented here can propose a number of salient image regions represented by pooled convolutional feature vectors. They used Faster R-CNN [38] in bottom-up attention. The top-down attention predicts an attention allocation over visual regions based on task-specific context. Finally, A weighted average is taken overall image features of all object regions and considered it as the attended feature vectors.

The Faster R-CNN [38] used in this work, had ResNet-101 [39] CNN as the backbone. The output of this approach was a set of image features that can be used in image captioning and visual question answering task. An IoU threshold was used to remove false and unimportant object classes from the output. From each detected object the region features were the mean-pooled convolutional feature having dimension 2048.

How F-RCNN detects objects and language decoder architecture are shown in figure 2.2. Their main contribution was to introduce two attention mechanism: bottom-up and top-down, in one architecture and to calculate attention at the level of objects and salient image regions. But the core language decoder is a recurrent network(LSTM) making the training and inference latency comparatively more than transformer-based architecture.

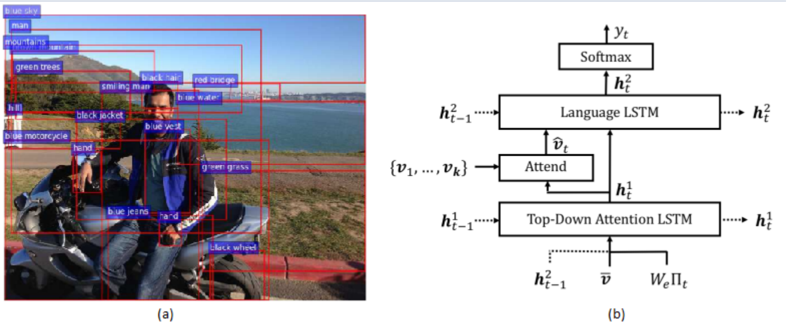


Figure 2.2: (a) F-RCNN Object Detections, (b) Language Decoder

2.7.2 Exploring Visual Relationship for Image Captioning [2]

In this work, Yao et al. presented a novel architecture of that time with Graph Convolution Network(GCN) and Long-Short Term Memory (LSTM). Their main motivation was to explore the visual relationship of objects shown in the image and boost the caption generation system. They explored two different types of visual relationships: Semantic object relationship and Spatial object relationship. To extract these relationship features they devised Graph Convolution on the region level representation of objects. Finally to generate captions a LSTM based language attention decoder is used using the extracted relation aware region level representation from GCN.

Semantic object relationship is expressed as <*subject-predicate-object*> for each pair of objects in the image. These relations are directional in a sense that it related two object by a predicate describing an action. To extract these relations they devised a classification model

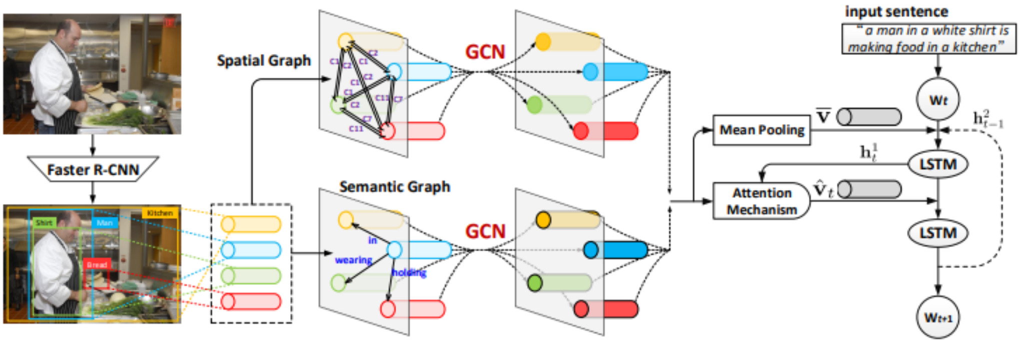


Figure 2.3: Graph Convolution plus Attention LSTM [13]

that takes detected object regions and predicts the predicate among the objects. They pre-trained this classification model on Visual Genom dataset.

Spatial object relationship is expressed as *object-object* for each pair of objects unexploiting their semantic relations. This relationship forms a spatial graph with edges *object-object* representing their relative geometric positions.

The context encoder of their framework is based on Graph convolution network which encodes information from the extracted semantic and spatial relationship graphs into a transformation matrix. Each entry of this matrix represents a relation-aware region-level feature. For caption generation purpose they devised an attention LSTM of 2 layers which takes the transformation matrix as input. Since LSTM is a class of Recurrent Neural network it process the transformation matrix sequentially and generates sentence word by word. A full overview of their architecture is shown in figure 2.3

Their works are tested on MSCOCO Caption dataset. But to train spatial relationship classifier they had to pre-train it on visual gnome. Finally they have shown their model performance using both machine evaluation on test server and human evaluation. At that time this work became the new state of the art on MSCOCO Caption task.

2.7.3 Self-Critical Sequence Training for Image Captioning [3]

In this work, Rennie et al. proposed a new approach to sequence training named as self-critical sequence training (SCST) and have shown that it dramatically improves image caption generation system of existing models. SCST is a class of reinforcement learning. It doesn't estimate the reward signal or normalize the reward signal. It utilizes the output of its own test-time inference to normalize the reward signal. Therefore, in case of imame captioning, only the outperforming words of sentence gets selected suppressing the inferior words.

This work is motivated from the study that existing sequence modeling systems like image captioning uses cross-entropy loss and evaluated with natural language processing metrics like BLUE [40], METEOR [41], CIDEr [42]. The problem with these metrics are everyone of the is non-differentiable making the system unable to directly optimize the metrics. Recent study have shown them that Reinforcement learning can solve this issue.

They made a baseline using the reinforcement learning using the inference algorithm of test time to get reward of the current model. So the loss function gradient with respect to time t is:

$$\frac{\delta L(\theta)}{\delta S_t} = (r(w^s) - r(\hat{w)})(P_\theta(w_t|h_t) - 1_{w_t^s})$$

where W^s is the negative reward of the current model, $r(\hat{w)}$ is reward from current model using the inference algorithm of test.

They experimented on MSCOCO [11] Caption dataset with a model consisting of spatial CNN features of attention models and Long-Short Term Memory as decoder. All their experiments were to optimize the CIDEr scores only. Before testing on the evaluation server they ensemble 4 variant of their approach and became the new state of the art of their time. They were able to successfully boost the CIDEr score from 104.9 to 114.7.

2.7.4 Boosting Image Captioning With Attributes [4]

In this work Yao et al. presented a novel architecture Long-Short Term Memory with Attribute (LSTM-A) for improving image captioning system. Their study shows that if higher level attributes of objects are incorporated with the captioning system then it generates more enhanced and accurate captions.

The full architecture can be divided into two parts: image feature extraction with object attribute selection and LSTM decoder. Machine cannot understand plain words directly. So at first they represented each word of captions using one hot vector. For image feature extraction they used the second last layer of pre-trained GoogleNet. They used top 1000 word of MSCOCO captions for attribute generation. Using the MILL [43] model they detected the possible attributes of images.

The caption generator is a special type of recurrent neural network known as LSTM. Existing models only used the extracted image features as input to the LSTM. Their main idea was to incorporate the extracted higher level object attributes with it. They tried with five different ways to incorporate both image and attribute to the LSTM. A diagram view is shown in figure 2.4.

The first variant is only injecting the attribute representation directly as a sequence to first time step of LSTM to inform the LSTM about higher level attributes. In the second variant

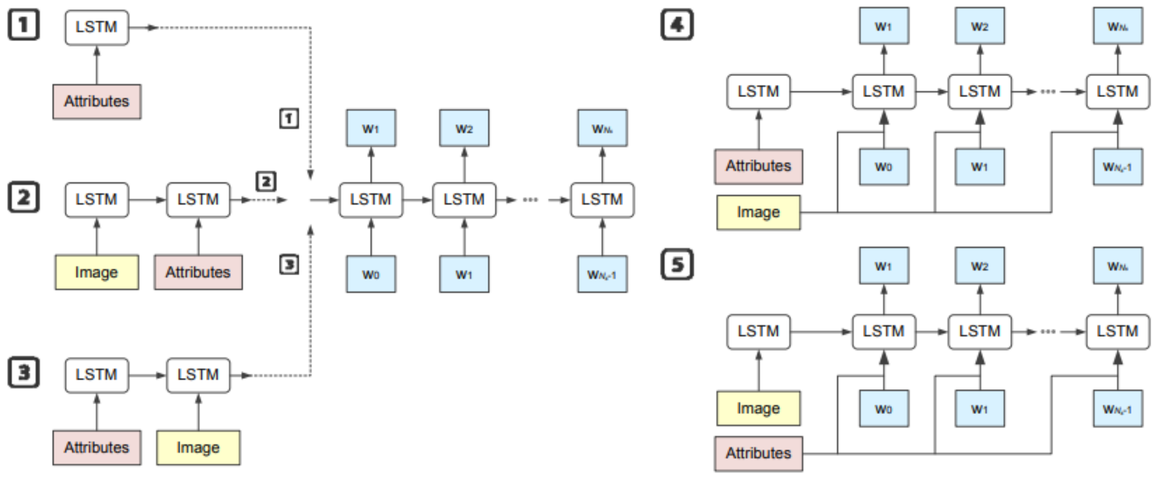


Figure 2.4: Five variants of LSTM-A framework

image features are injected into the first time step of the LSTM and attribute representations are injected into the second time step of LSTM. On the third variation attribute representation was at the first time step and image features were at the second time step of LSTM only. The fourth variation uses the image features on every subsequent time steps of the LSTM leaving attribute representation on the first time step. The fifth variation just swapped the image and attribute injection of the fourth variation.

This experiment was also done on MSCOCO Caption dataset. Their caption model was implemented based on Caffe which is a popular learning framework. The sentence generation has two ways: beam search and selecting the most probable word. They used the beam search technique which is to select top k candidate word and use them for next word generation. In all of their experiment the value of $k = 3$. Finally they checked their model performance on the evaluation server of MSCOCO and compared with existing models like NIC [33], LRCN [44], Hard Attention & soft attention [45]. The overall performance on MSCOCO at their time was better than those models.

2.7.5 Captioning Images Taken by People Who Are Blind [5]

Danna Gurari et al. [5] were first to introduce a dataset specially for visually impaired people of image captioning. Main motivaiton was to create a dataset where all the images will be captured by visually impaired people. This new dataset, which we call VizWizCaptions, consists of over 39,000 images. The dataset is publicly available at https://vizwiz.org.Each image in this dataset has 5 captions obtained by crowd-sourcing. Some examples can be seen in figure 2.5.

The images of this dataset were build on their two existing datasets of image taken by real user of visual description service. Each images were taken by user's mobile devices with optional voice recorded question. The caption collection were done through crowd-sourcing of Amazon Mechanical Turk(AMT). The crowd-workers were instructed to create caption with all objects that may be important for visually impaired people. Since the image quality was not good for a large portion of the images, the crowd-workers had a bias to create captions by just saying too sever quality to make captions. That's why they were also instructed if possible avoid this kind of caption generation. After some post processing they published the train, validation split with ground truth captions and test split without the ground truth captions. In order to evaluate captioning system they hosted a evaluation server on eval.ai for test split.

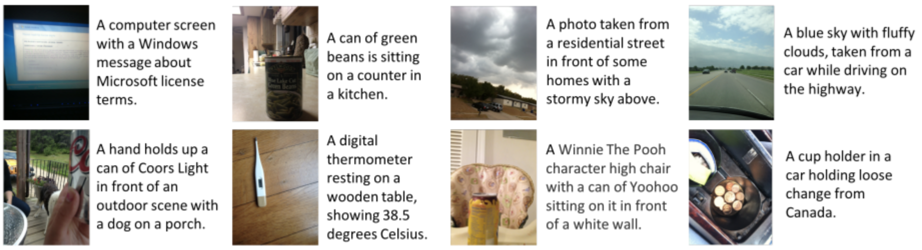

A computer screen with a Windows message about Microsoft license terms.
A can of green beans is sitting on a counter in a kitchen.
A photo taken from a residential street in front of some homes with a stormy sky above.
A blue sky with fluffy clouds, taken from a car while driving on the highway.
A hand holds up a can of Coors Light in front of an outdoor scene with a dog on a porch.
A digital thermometer resting on a wooden table, showing 38.5 degrees Celsius.
A Winnie The Pooh character high chair with a can of Yoohoo sitting on it in front of a white wall.
A cup holder in a car holding loose change from Canada.

Figure 2.5: Examples of VizWiz Captions

2.7.6 Image Captioning as an Assistive Technology: Lessons Learned from VizWiz 2020 Challenge [6]

In this work, Dongin et al. presented their contribution on image captioning task for visually impaired people focusing on their winning model of VizWiz 2020 Challenge. This caption model is basically a multi-modal Transformer. It consists of an image feature extractor, OCR token extractor, Object detector, an encoder and a decoder. The model takes an image, generates 3 modality streams of image features, detected object labels and OCR results. Detected object labels and OCR tokens are passed through a word embedding layer known as FastText. Then the embedding and the extracted image features are concatenated to form the input of the transformer encoder. The encoded context of from the encoder then passed through the multi modal transformer decoder. For CIDEr optimization the output of decoder is passed through the SCST [3]. During this phase, a copy mechanism is introduced to reflect the extracted OCR token into the generated captions. A post processing layer refurnishes the final caption.

Image Feature extractor is built on ResNeXt model extracts the Image features. The faster-RCNN seqs detects object labels (10 max) and a Open Source OCR module extracts plain text word (20 max). The inner mechanism of their Dictionary-Guided Rotation-Invariant OCR module is shown in figure 2.6. By rotating the image 4 times 4 sets of OCR tokens are generated in the OCR module. The rotation that gives a large number of meaningful OCR token is selected.

An overview of their multi-modal assistive caption is given in figure 2.6.

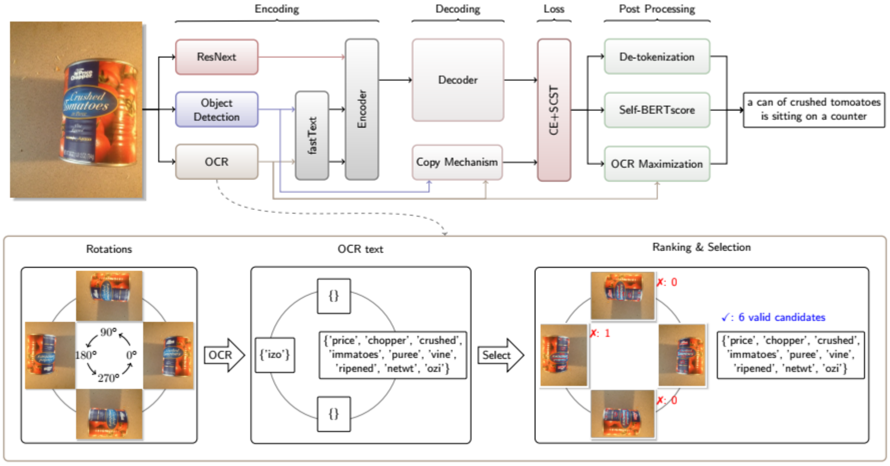


Figure 2.6: Multimodal Assistive Captioner [13]

This work was one of the pioneer for using plain text from the image in the generated captions for image captioning. Instead of letting the model pick OCR tokens randomly from detected tokens, they used smart copy mechanism to use them as it is. Beside that, unlike other general models they didn't only use the cross-entropy loss. To optimize CIDEr score they used SCST [3] along with the cross-entropy loss. Still model has some limitations like incorporation of OCR tokens introduced a problem of copying irrelevant text from the image and multi-word names were broken.

2.7.7 An Improved Feature Extraction Approach to Image Captioning for Visually Impaired People [7]

This work extended from actual AoA-Net [9]. *Dong Wook Kim et al.* [7] modified the current AoA-Net refining module by placing a focusing module between multi-head attention module and AoA module, as shown in Figure 2.7. Each focusing module has gated linear unit(GLU). These GLUes can learn relationships of individual objects with other objects. At first image features A are extracted using faster-RCNN [38] and passed through AOA encoder. The transformation is done as follow:

$$A' = LayerNorm(A + AoA(f_{mh-att}, W^Q A, W^K A, W^V A))$$

where W^Q, W^K, W^V are weight matrics and f_{mh-att} is the multi-head attention function.

A graph convolution operations is applied on feature map A' which is the output from the refining module. This graph convolution encodes regions with visual relations both in spatial

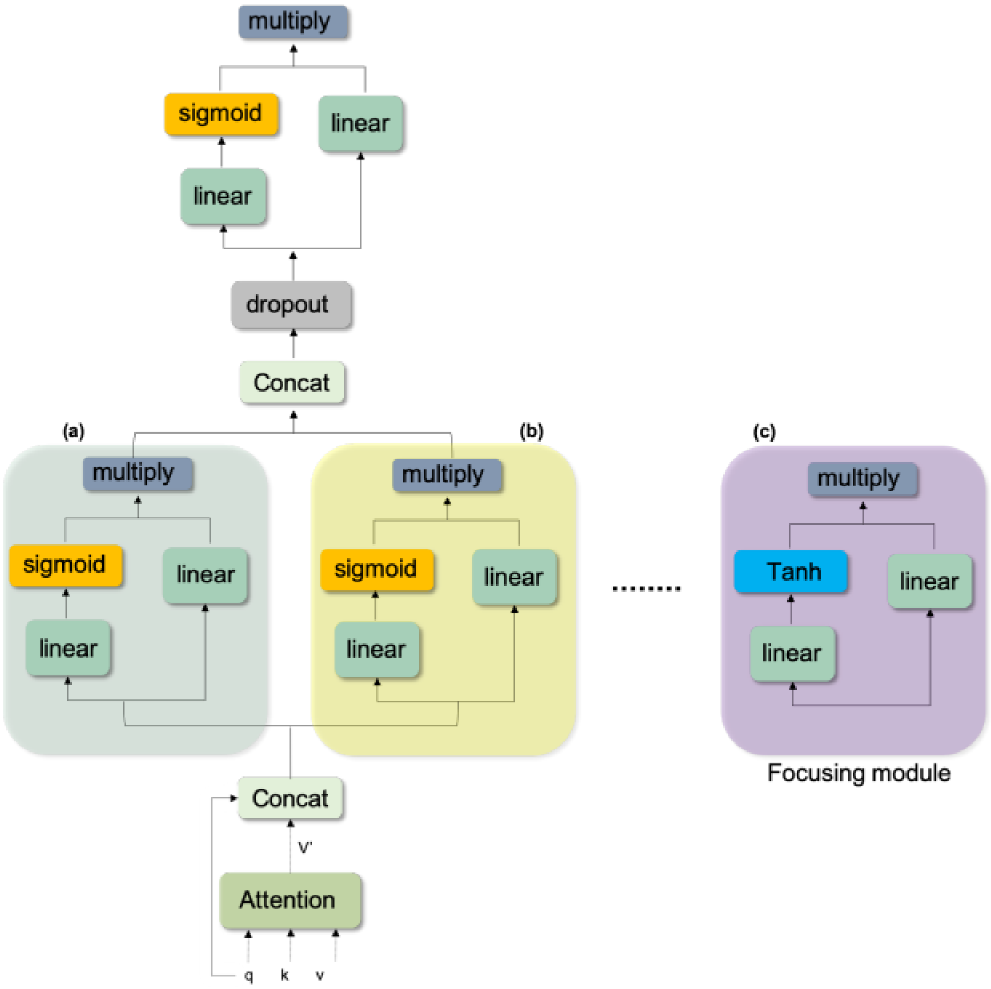


Figure 2.7: Proposed encoder refining module includes GLU based focusing module.

and channel-wise domains enabling the model to extract global contextual information from incomplete images. For language decoder they used pre-trained language model PLM. This pre-trained language model (PLM) generates embedding table for each words which helps in extracting fine grained text information. The full model architecture in shown in figure 2.8. This architecture is tested on VizWiz-Captions [5] dataset. Their resulting score crossed the actual AoA-Net by a very small margin in every evaluation metrics shown in AoA test result.

2.7.8 ReFormer: The Relational Transformer for Image Captioning [8]

In this work, *Xuewen Yang et al.* [8] proposed a novel architecture named ReFormer- a RElational transFORMER that can generate image features with object to object relational information. The encoder part of that model can explicitly show object to object relationships in the image by plain words. The current image captioning models are highly depended on pre-trained models rather than encoder its self. From this motivation, they came up this idea that have only

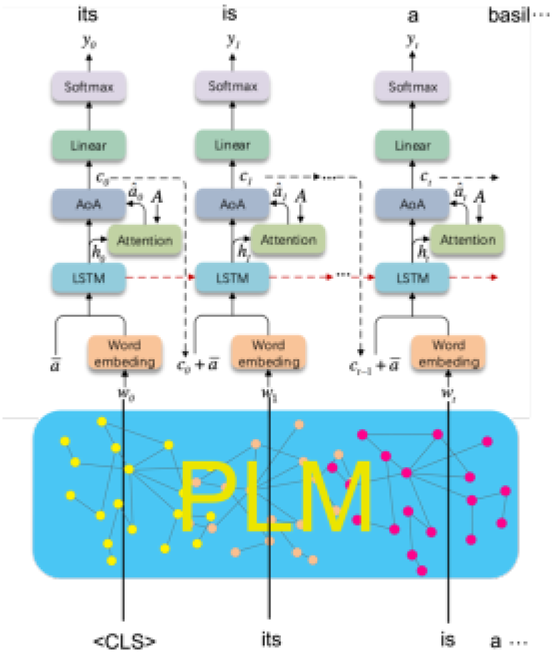


Figure 2.8: Proposed decoder module with PLM word embedding

a encoder and decoder making the system more flexible. The full model architecture is shown in figure 2.9.

The model architecture can be divided into two parts: Relational Encoder and weighted Decoder for caption generation. The relational Encoder has a pre-processing layer followed by vanilla transformer [21] encoder. The encoder takes the pre-extracted bounding box of objects and their region features and returns a encoding context.

To exploit the pair-wise object relationship they trained the encoder first with a post processing layer on visual genome for the task of scene graph generation. During this training phase, the encoder learned the object to object predicates. The post processing layer was responsible for the predicate classification. The weighted decoder layer is also almost same as vanilla transformer decoder.

They experimented with their architecture on MSCOCO Caption dataset on karpathy split. For scene graph generation task experiments were done on Visual Gneom [22] dataset. They claimed their novel architecture have outperformed the current state of the art on MSCOCO and visual gnome.

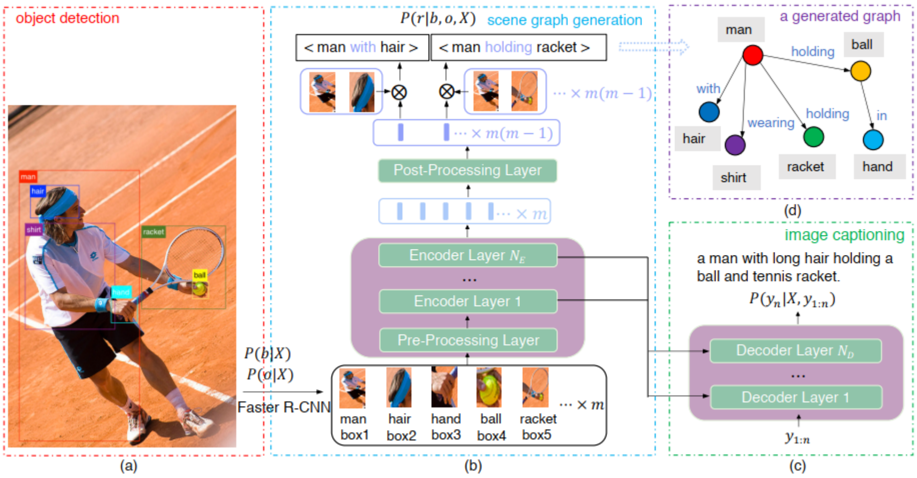


Figure 2.9: (a) Faster R-CNN provides m bounding boxes, object labels and object region features (b) Transformer encoder to generate m × (m - 1) pair-wise relations (c) Transformer decoder for generating captions. (d) The generated scene graph.

Chapter 3

System Architecture

We aimed to fix the research gap mentioned earlier in our study. Our idea is to fusion existing state of the art models to make a pipe-line that can solve some of those issues. Our captioning pipeline contains a scene graph generator, global feature extractor, a OCR token extractor and an encoder-decoder framework. Each of these module is responsible for solving our research objectives. An overview of our proposing pipeline is shown in figure 3.1.

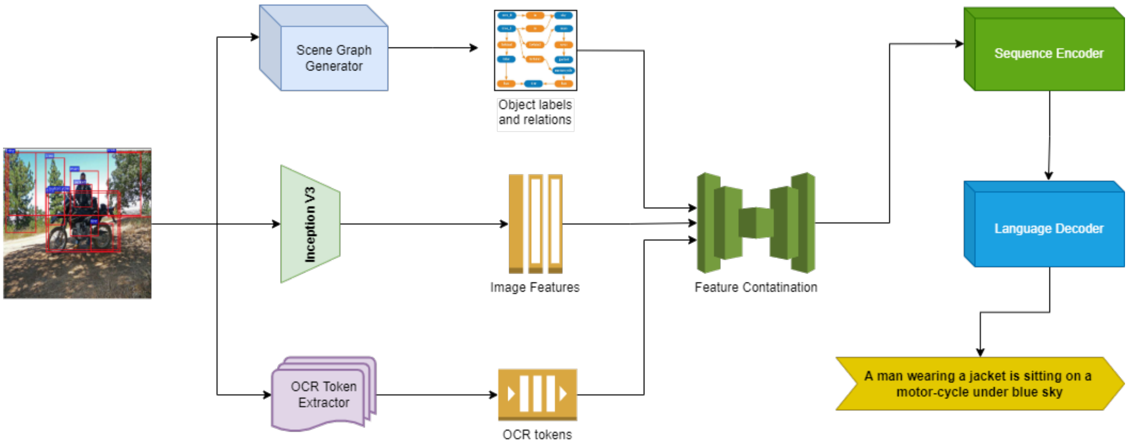


Figure 3.1: Scene Graph and Language decoder based Image Captioning

3.1 Model Description

This pipeline can divided into 5 module:

- Pre-trained Scene Graph Generator.
- Pre-trained Global Feature Extractor.
- OCR Token Extractor.
- Sequence Encoder.

- Language Decoder.

3.1.1 Scene Graph Generator Module

Since our working dataset doesn't have any ground truth scene graphs, we used pre-trained state of the art scene graph generation model introduced by Kaihua Tang et.al [14]. This model is currently on the top to generate scene graphs from images trained on visual gnome one of the larges dataset for scene graph generation. A overview of their model is shown in figure 3.2

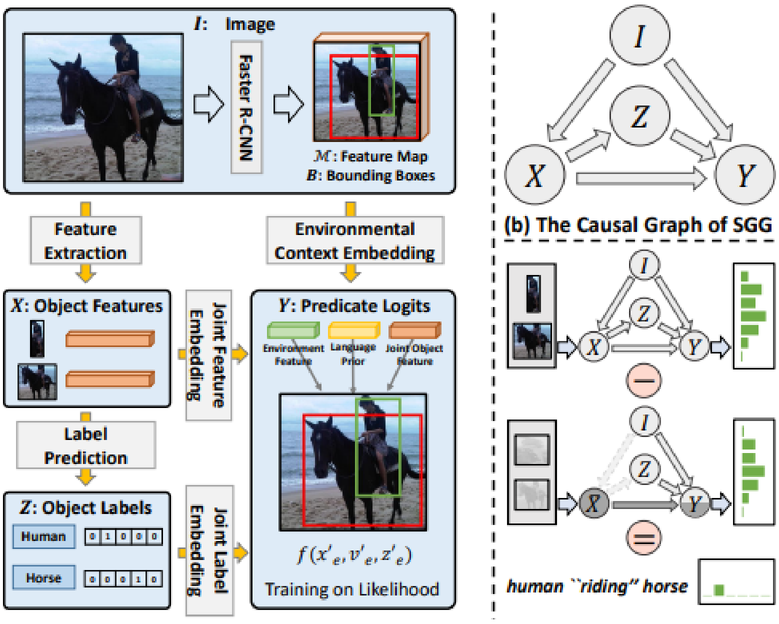


Figure 3.2: Scene Graph Generator Pipeline, given by Kaihuan Tang et.al [14]

The SGG framework shown in figure 3.2 works like a Causal Graph [46]. A causal graph is a graph $G(N, E)$ where a set of nodes N interact among themselves by a set of directed connections E. If the nodes represents objects in the image then the graph provides us the causal relationships between every pair of objects. In sub figure (b) of figure 3.2 we have three variable I, X, Z, Y having directed connection. I represents input image, X represents object feature, Z represents object class, Y represents SGG predicate.

The model takes raw image as input. A pre-trained faster R-CNN [38] detects object and returns each objects feature map M and their bounding boxes $B = \{b_i | i = 1, 2, 3\}$ from the input image I. In link $I \rightarrow X$ object features are extracted from RoiAligh feature r_i, the bounding box b_i and tentative object labels l_i by the faster-RCNN.

$$Input : \{r_i, b_i, l_i\} \Longrightarrow Output : \{x_i\}$$

From each objects feature map object labels are predicted first which is represented by the link

$X \to Z$. This process is the object classification. The labels are then decoded from corresponding x_i features.

$$Input : \{x_i\} \Longrightarrow Output : \{z_i\}$$

After that, through special environmental embedding, joint feature map of each box and their label, each object to each objects relations are predicted. Here it is known as predicate classification. This classification is done in node Y.

The link $X \to Y$ represents the Object feature Input for the SGG. In this process the pairwise feature X are merged into joint representation for relationship classifications. The link $Z \to Y$ represents the Object class input to the SGG. Each object class is plain text. So before passing to SGG it is transformed into word embedding by a joint embedding layer.

$$z'_e = W_z[z_i \otimes z_j]$$

where $\otimes$ is responsible for generating one hot vector representation of the object label.

The link $I \to Y$ represents the global visual context input to the SGG. The node Y incorporates all these 3 branch inputs by a fusion function and objected relationships i.e predicate classification is done. There 2 fusion functions, used by Kaihuan Tang et.al [14].

$$SUM: \; y_e = (W_x x'_e + W_v v'_e + z'_e)$$

$$Gate: \; y_e = W_r x'_e \dot{\sigma}(W_x x'_e + W_v v'_e + z'_e)$$

The final output consist of object labels from node Z, object bounding boxes from node X and pair wise object relationships from node Y.

Our focused dataset (VizWiz-Captions) doesn't have ground truth predicates. So training this SGG is not possible on VizWiz-Captions. That's why we used pre-trained(on Visual Gnome [22]) version of this model. From this pre-trained Scene Graph Generator we get the grounding objects labels and their relationship labels to each other which is further passed through a embedding layer to convert them into feature vectors. Finally concatenated with other features. We expected that this module will overcome the research gap of "Missing out object relation in the generated caption" and "Missing of partially visual objects".

3.1.2 Global Feature Extractor Module

In order to keep the semantic information and visual grounding along with object relationship we need overall picture of the image. From this idea we kept a Global feature extractor module. The backbone of this feature extractor is InceptionV3 [20] network which is pre-trained on ImageNet dataset. To have the feature vectors we had to discard last classification layer of inceptionV3. Now it takes the raw image and returns a (64 x 2048) dimensional feature vectors. These features in further concatenated with object relation features. The inceptionV3 architecture is shown in figure 3.3.

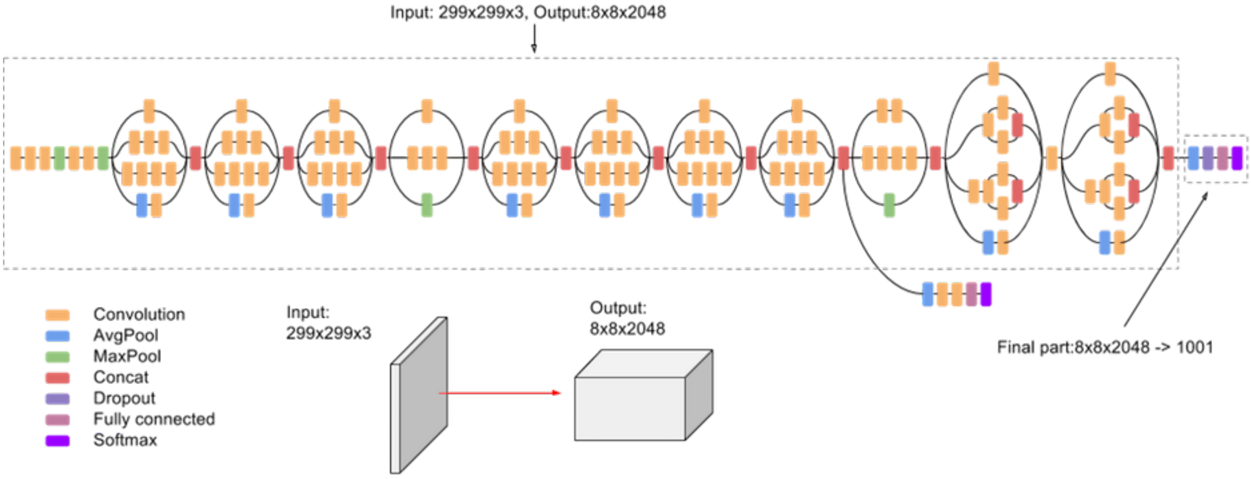


Figure 3.3: Inception-V3 pre-trained on ImageNet

The inceptionV3 has still the core of inceptionV1. The whole model can be broken into small inception module. Each module consists of 3 layers with 4 parallel pipe. It has 3 convolution layers of size 1x1, 3x3 and 5x5 respectively and 1 max pooling layer of size 3x3. A zoomed in version of inception modules is shown in figure 3.4.

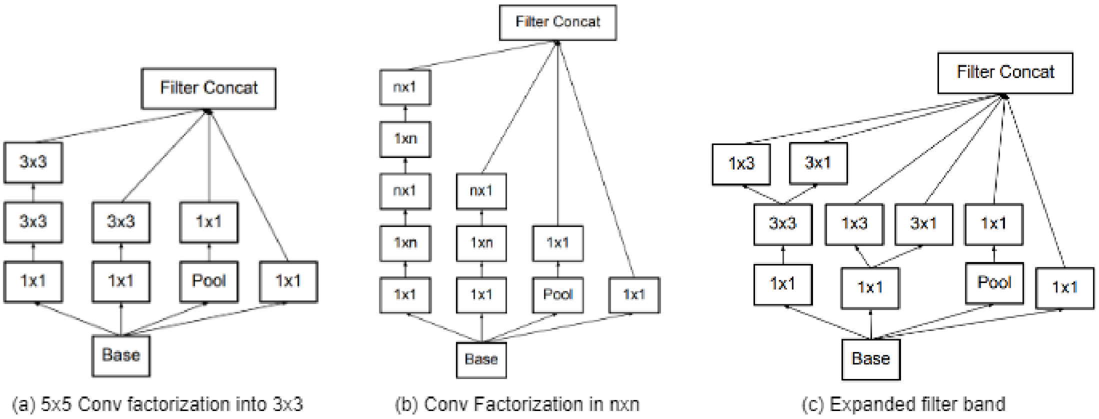


Figure 3.4: Inception Module of inceptionv3

The full model dissection is shown in table 3.1 We don't need the classification layer. So we discarded the last classification layer(pool, linear and softmax) to get the higher level feature vectors. So our Global feature extractor is the inceptionV3 model up to it's last inception layer that returns a feature map of $(8x8x2048)$ dimension. The sequence encoder only takes 2 dimensional features. For this reason we resized the $(8x8x2048)$ dimensional features into $(64x2048)$ dimensional features. This final features is concatenated with the scene graph features and OCR token embedding to form the final context metrics for sequence encoder.

Type	Patch size/stride	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in figure 3.4 (a)	35×35×288
5×Inception	As in figure 3.4 (b)	17×17×768
2×Inception	As in figure 3.4 (c)	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

Table 3.1: InceptionV3 Layers

3.1.3 OCR Token Extractor

Since many of the images contains plain text, generated captions should reflect them. The ground truth captions also contains them. In order to extract those text, we incorporated a open-source OCR token extractor within our pipeline. The used module is Paddle [47] OCR. Each extracted tokens are then passed through a Glove [25] embedding layer to convert it into

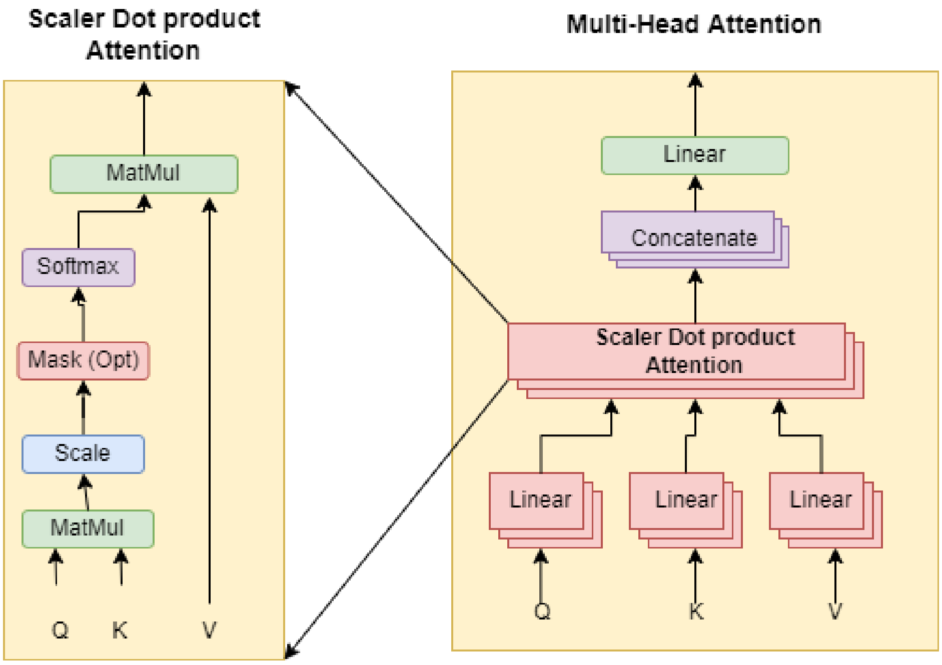


Figure 3.5: Transformer Scalar dot product and Multi-head attention

a feature vector of 300 dimensions. But our sequence encoder needs 2048 dimensional input. To transform 300 dimensional feature, each embedding then linearly projected on 2048 dimensional feature vectors.

$$e^{300} = Glove(OCR\ token : {}'food')$$

$$e^{2048} = Linear(e^{300})$$

Finally each 2048 dimensional feature vectors are concatenated with the scene graph and global features and passed to the Sequence encoder.

3.1.4 Sequence Encoder

The output of the above mentioned three module are concatenated to form a (N X 2048) dimensional features. We think them as a sequence of N features. This sequence is passed through the Sequence encoder. Sequence encoder is nothing but the vanilla encoder part of the transformer [21] network. Each of the 2048 dimensional features works as the word embedding vector in the encoder part. Using 8 head of multi-head attention and feed-forward network these features are encoded as context which is then passed to the language decoder.

The core module of Transformer encoder-decoder is scalar dot product and multi-head attention. How they are related is shown in figure 3.5. Multi-head attention is a part of Sequence encoder shown in figure 3.6. There is 6 encoder stacked in the sequence encoder. This sequence encoder takes the I^{Nx2048} features and each I_i^{2048} is copied 3 times, labeled as Q, K, V. These Q, K, V is passed through the multi-head attentions of each encoder of sequence encoder. The output of multi-head attentions are concatenated further and passed through a feed-forward network. The scalar dot product attention is calculated as:

$$Attention(Q, K, V) = softmax(\frac{Q\dot{K}^T}{\sqrt{d_k}})V$$

where d_k is the model dimension. In our pipe-line as well as the actual transformer network have $d_k = 512$. The multi-head attention is calculated by:

$$Multihead(Q, K, V) = Concat(h_1, h_2, ..., h_n)W^o$$

$$h_i = Attention(QW_i^q, KW_i^k, VW_i^v)$$

where W_i is weight metrics of each head. Passing through layer norm the multi-head attentions is sent to the feed forward network. After passing N=6 encoder layer our context is ready to be decoded by Language decoder.

3.1.5 Language Decoder

Language Decoder is also nothing but the Decoder part of vanilla transformer [21] network. Decoder takes the context encoding and partially generated caption. On each iteration Decoder uses 6 layer of 8 head of multi-head attention to find the relevant next word for the caption. The partially generated caption then again feed to the decoder to predict next word and it goes one until a special end token is generated. The architecture of Encoder Decoder is shown in figure [3.6].

The Language decoder is also a stack of N = 6 identical decoder layers. Unlike the encoder layer, the decoder has a sub-layer of multi-head attention between the first multi-head attention and the feed-forward network. Therefore a single decoder can be broken into 3 sub-layers:

- The first sub-layer gets the decoder stack's previous output, adds positional information, and applies multi-head self-attention to it. While the encoder is meant to pay attention to all parts of sequence in the input features, regardless of their location, the decoder is adjusted to only pay attention to the words that come before them. As a result, the prediction for a word at position can only be based on the known outputs for the words in the sequence before it. This is accomplished by introducing a mask over the values produced by the scaled multiplication of matrices and in the multi-head attention mechanism (which

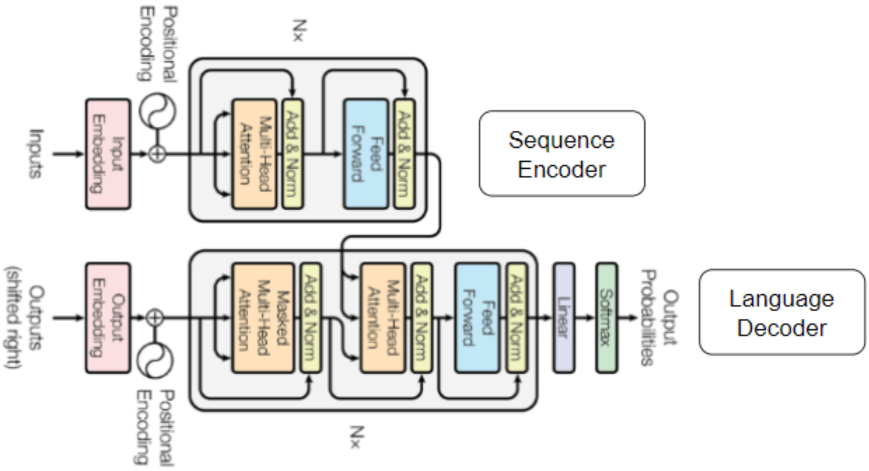


Figure 3.6: Transformer as Sequence Encoder and Language Decoder

performs numerous, single attention functions in concurrently).

- The second layer of the decoder utilizes a multi-head self-attention technique identical to the one used in the encoder. This multi-head mechanism gets the queries from the preceding decoder -, as well as the keys and values from the encoder output i.e context. The decoder can now focus on all of the parts of the input sequence.

- A fully connected feed-forward network, similar to the one established in the encoder's second sub-layer, is implemented in the third layer.

On the decoder side, the three sub-layers have residual connections around them and are followed by a normalization layer. In the same way that positional encodings were added to the encoder's input embeddings, positional encodings are added to the decoder's input embeddings.

Chapter 4

Results Analysis and Discussion

To find the performance and comparison of our proposed pipe-line we experimented our pipe-line on different setup.

4.1 Experimental Setups

4.1.1 Datasets

During background studies we faced many image captioning datasets. Most of them were consist of fine-grained images. In the following sub-sections there is a overview of our explored dataset. Among them we have chosed VizWiz-Caption dataset because it contains visually challenged images as they are taken by blind people.

Microsoft COCO Captions [11]

Microsoft COCO Caption dataset and evaluation server was first introduced in 2015 in a paper called "Microsoft coco captions: Data collection and evaluation server". The dataset contains 330,000 images of which 200,000 are labelled. Here we have more than 1.5 million object instances which can be classified into 80 object categories and 91 stuff categories. All images are well focused and good quality. Each image has 5 captions that were generated by crowd-sourcing. It has a well known karpathy train, validation test splits. To get test results a evaluation server formed. That test splits ground truth captions are hidden on the test server. For testing the candidate captions generated by captioning algorithm are uploaded to that server and the server return popular evaluation score of evaluation metrics like BLEU, METEOR, ROUGE and CIDEr. Some notable characteristics:

- 330K images (>200K labeled)

- 1.5 million visible object instances
- 80 object classes
- 91 stuff classes
- 5 captions per image
- All images are well-focused
- Captions are generated by crowd-sourcing

Flickr30k [18]

Flickr30k dataset was first introduced in 2014 in a paper called "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. Transactions of the Association for Computational Linguistics" [18]. It has over 30,000 images and over 150,000 captions. The captions were generated using crowd-sourcing. All the images are well focused and have good quality. Each image has 5 captions associated with them. Some notable characteristics:

- 31,783 images
- 158,915 crowd-sourced captions
- 5 captions per image
- All images are well-focused

VizWiz-Captions [5]

VizWiz-Captions dataset was first introduced in 2020 in a paper called "Captioning Images Taken by People Who Are Blind" [5]. This is the first dataset dedicated to image captioning for visually impaired people. It has over 38,000 images all taken by visually impaired people. For each image there are 4-5 captions associated with them. The captions were generated by crowd-sourcing. As the images are all taken by visually impaired people some of the images are not well focused, some of the images are blurry, some contain uncaptionable objects. Most of the images are taken indoor without proper lighting. Some notable characteristics:

- 23,431 training images
- 117,155 training captions

- 7,750 validation images
- 38,750 validation captions
- 8,000 test images
- 40,000 test captions
- Test annotations/captions are hidden.
- Test Scores are found on evaluation server

BanglaLekhaImageCaptions [48]

The first Bengali image captioning dataset is BanglaLekhaImageCaptions. It was initially mentioned in a study titled "Chittron: An Automatic Bangla Image Captioning System"cite4 in the year 2020. This dataset is solely for image captioning in Bengali. There are 9,154 images in total, each with two captions. Two native Bengali speakers are responsible for the captions. There is also a lot of human bias in the dataset. This bias makes it difficult for any model to describe nonhuman subjects. There were 7154 images used for training, 1000 for validation, and 1000 for testing. Some notable characteristics:

- 7154 training images
- 1000 validation images
- 1000 test images
- Only two captions per image
- Human object bias
- First dataset in Bengal geo-culture

ROCO [49]

Radiology Objects in Context or ROCO is a dataset of medical images from different medical domain. It was first published in 2018 at International Workshop on Large-scale Annotation of Biomedical data and Expert Label Synthesis. It contains 81,000 radiology images. Images were obtained by CT Scan,X-Ray, Ultrasound Imaging, Fluoroscopy, Positron Emission Tomography(PET), Mammography, Magnetic Resonance Imaging(MRI), Angiogram etc. All the data points consist of a image, single or multiple captions, keywords, CUI Code and Semantic Type Code and label. There is also an out-of-class set containing 6k images. Each of those image are based on synthetic radiology figures and digital arts.

4.1.2 Evaluation Metrics

Machine learning models depend on constructive feedback systems. Normally while training the model, takes feedback from a system and makes improvements based on that. The aspect of this is to discriminate between two models and see which one is performing better. So normally as our model is machine learning based, it uses some evaluation metrics too.

Most of the evaluation metrics that are used in Image captioning tasks are string similarity based. Say, given an image and your model generates a caption, you'll check similarity of your captions with some predefined captions and give feedback based on that similarity. Some popular evaluation metrics used in image captioning tasks are described bellow:

Bilingual Evaluation Understudy (BLEU) [40]

BLEU calculates scores based on the similarity between two given sentences. The similarity depends on number of segments that match between them. Based on the size of the segment BLEU can be divided into some sub metrics too. The score is calculated by:

$$P_n = \frac{\sum\limits_{C \,\epsilon\, \{candidate\}} \sum\limits_{n-gram \,\epsilon\, C} Count_{clip}(n-gram)}{\sum\limits_{C' \,\epsilon\, \{candidate\}} \sum\limits_{n-gram' \,\epsilon\, C'} Count_{clip}(n-gram')}$$

- BLEU-1: Segments of size 1 are calculated for matches.
- BLEU-2: Consecutive segments of size 2 are calculated for matches.
- BLEU-3: Consecutive segments of size 3 are calculated for matches.
- BLEU-4: Consecutive segments of size 4 are calculated for matches.
- BLEU-n: Consecutive segments of size n are calculated for matches.

Recall Oriented Understudy of Gisting Evaluation (ROUGE) [50]

ROUGE is actually a metric for summarising of text. But it is also used for evaluation of machine translation. It takes the generated summary or translation and a set of candidate summaries or translations (human annotated) and returns a score on how much they correlate. The score is calculated by:

$$ROUGE-N = \frac{\sum\limits_{S \,\epsilon\, \{Ref.Summaries\}} \sum\limits_{n-gram \,\epsilon\, S} Count_{match}(n-gram)}{\sum\limits_{S \,\epsilon\, \{Ref.Summaries\}} \sum\limits_{n-gram \,\epsilon\, S} Count(n-gram)}$$

It has so far five variant for evaluation.

- ROUGE-N: Find the frequency of overlapping N-grams between the generated summary and candidate summaries.
  - ROUGE-1 considers uni-gram (each word) overlapping of the generated summary and candidate summaries.
  - ROUGE-2 considers the bi-gram overlapping of the generated summary and candidate summaries.
- ROUGE-L: It is based on Longest Common Subsequence (LCS). Therefore it can understand sentence level similarity and can find the longest co-occurring n-grams sequence.
- ROUGE-W: It is based on weighted LCS which can find consecutive LCSes .
- ROUGE-S: It is based on Skip bi-gram co-occurrence.
- ROUGE-SU: It is a mixure of Skip-bigram and unigram co-occurrence.

Metric for Evaluation of Translation with Explicit Ordering (METEOR) [41]

METEOR is calculated based on recall and precision. By looking at the uni-grams precision, it combines recall and precision as weighted scores. Then it recalls and aligns the generated output with each ground truth candidates individually and takes the best score from each pair of output and ground truth candidates. It is calculated by the following equations:

$$P = \frac{m}{w_t}$$

$$R = \frac{m}{w_r}$$

$$F_{mean} = \frac{10PR}{R + 9P}$$

Where P is precision, R is Recall, m is number of uni-grams in candidate sentences found in reference sentences, w_t number of uni-grams of candidate sentence, w_r number of uni-grams of reference sentences. F_{mean} is the harmonic mean combined with precision and recall. Through word inflection variation, paraphrasing match, interchangeable words it can overcome translation variability. This enables the metric to find similarity over semantic equivalency. Beside, METEOR can keep fluency by directly penalizing the word order. This way METEOR can find better correlation with ground truth sentences.

Consensus-based Image Description Evaluation (CIDEr) [42]

Consensus-based Image Description Evaluation (CIDEr) is the measurement of how much a generated caption is similar to a set of ground-truth sentences.It tries to solve the problem of weak correlation among the previous metrics like Blue and the judgments of human. CIDEr however shows higher correlation with consensus and cognitive judgements by humans. It captures the sentence similarity, grammatical correctness, sentence salience, and accuracy based on precision and recall.

$CIDEr_n$ score for n-grams of length n is calculated by the mean cosine similarity between the candidate sentence and the reference sentences. It needs both the recall and the precision. The equations are:

$$CIDEr_n(c_i, S_i) = \frac{1}{m}\sum_j \frac{g^n(c_i).g^n S_i}{||g^n(c_i)||.||g^n S_i||}$$

where $g^n(S_i)$ is a vector formed by $g_k(S_i)$ corresponding to all n-grams of length n and $||g^n(S_i)||$ is the magnitude. $g^n(c_i)$ represents similarly. Scores from n-grams of varying lengths are calculated by:

$$CIDEr(c_i, S_i) = \sum_{n=1}^{N} w_n CIDEr_n(c_i, S_i)$$

,

Semantic Propositional Image Caption Evaluation(SPICE) [51]

Semantic Propositional Image Caption Evaluation or SPICE is an evaluation metric where comparison is made based on multiple captions instead of single one. All the existing metrics work based one n-gram similarity which can produce false negative results as n-gram similarity might not ensure similar caption. SPICE intends to reduce the number of false positives. Even though it tries to reduce false positives, it does not however reduce any false negative results. The equations for SPICE metrics are:

$$G(c) = \{O(c), E(c), K(c)\}$$

$$T(G(c)) \triangleq O(c) \cup E(c) \cup K(c)$$

$$P(c, S) = \frac{|T(G(c)) \otimes T(G(S))||}{T(G(c))|}$$

$$R(c, S) = \frac{|T(G(c)) \otimes T(G(S))|}{|T(G(S))||}$$

$$SPICE(c, S) = F_1(c, S) = \frac{2\dot{P}(c, S)\dot{R}(c, S)}{P(c, S) + R(c, S)}$$

Here $O(c)$ is set of objects in caption c, $E(c)$ relational edge for objects in caption, $K(c)$ is the set of attributes related to caption c. $G(c)$ represents the parsing scene graph. $T(G(c))$ is nothing but a logical tuple of parsing graph. $P(c, S)$ and $R(c, S)$ are precision and recall respectively. Spice score is actually the F_1 score from precision and recall.

The key components of SPICE are metric formulation and comparison considering multiple captions. A scene graph is generated from all the available captions and after that SPICE score is calculated as an F-score over tuples. The matching tuples in the captions use WordNet synonym approach to ensure that tuples are matched even if the exact form is not given but lemmatized forms are present. It only considers full match so no partial match gets any score even if some elements in the tuple is correct. This is considered a better evaluation metric as it considers similarity attributes and relations along with the exact wording for score calculation.

4.2 Experiments

To find the impact of each module in our proposed pipeline we experimented on different setup. Each setup is described below.

4.2.1 Without Scene Graph Generator

In this setup we further have two different training with OCR token and with out OCR tokens. From our pipeline we removed the scene graph generator. At first we used the pre-trained InceptionV3 network to extract the global features and cached them. The output dimension of inceptionV3 was (8 x 8 x 2048). We resized them into (64 x 2048) and treated them as a sequence of 64 tokens with 2048 dimensional embedding. These sequence are then feed into the the sequence encoder. Each ground truth captions were encompassed with special <start> and <end> token to denote start and end of the captions.

On each iteration a prefix of ground truth caption and context encoding was feed into the language decoder. Language decoder predicted the next word. Loss was calculated using sparse categorical loss.

In case of OCR token, we had pre-extracted OCR token for the images by another author. First we tokenized the OCR token with a tokenizer to get their corresponding integer representation. Then tokens are formed into a sequence and passed through a embedding layer.

	Without OCR Token	With OCR Token
Dataset	VizWiz Caption	VizWiz Caption
Batch Size	64	64
train epoch	20	10
Data point	155k	40K
Train Loss	1.1661	0.8312
Accuracy	0.2077	0.3135
System	Google Colab	Google Colab

Table 4.1: Environmental setups for experiment excluding Scene Graph

Embedding layer gave us a feature vector of 2048 dimension. It was then concatenated with the global features forming a sequence of 128 feature vectors. The environmental setups of these two approaches are shown in table 4.1.

Results

Corresponding scores are shown in table 4.2. We can see that use of OCR token gives a boost in resulting score.

Model	B-1	B-2	B-3	B-4	METEOR	ROUGE-L	CIDEr	SPICE
InceptionV3+ Transformer	31.04	16.51	8.57	4.43	8.27	20.97	14.44	5.69
InceptionV3+ OCR+ Transformer	40.06	22.06	12.22	6.42	11.75	29.11	13.68	5.86

Table 4.2: Scores without using Scene Graph Generator [B-n refers to BLEU-n]

4.2.2 With Scene Graph Generator

In this setup we also further have two different training with OCR token and with out OCR tokens. Here we used the previously cached global features generated by inceptionV3.

In order to get the object label and their relation labels we used the pre-trained Scene Graph Generator given by Kaihuan Tang et.al with pre-trained weight of visual gnome. It was a lengthy process and google colab had limited memory. That's why we could not generate scene graphs for each of the training and validation split. We only generated scene graph for 16000 images using validation and test split. Each scene graph gave us a set of object labels and relation labels among the objects. We saved them into a json file. The whole process took 6 hours.

After that each object labels and relation labels are passed through a embedding to convert them into a feature vectors of 2048 dimensions and tokenizer to get their corresponding token number. Those features are then concatenated with global features forming a sequence of 128

Type	Without OCR Token	With OCR Token
Dataset	VizWiz Caption	VizWiz Caption
Batch Size	64	64
train epoch	20	20
Data point	40k	40K
Train Loss	3.916	3.035
Accuracy	0.0217	0.0298
System	Google Colab	Google Colab

Table 4.3: Environmental setups for experiment including Scene Graph

feature vectors. In case of OCR token incorporation, we followed the method mentioned in the previous experiment. The environmental setups of these two approaches are shown in table 4.3.

Results

Corresponding scores are shown in table 4.4. We can see that use of OCR token increased the resulting score. But overall result was very disappointing. We further explored the possible reasons of this kind of unexpected behaviour. Findings are shown in later section.

Model	B-1	B-2	B-3	B-4	METEOR	ROUGE-L	CIDEr	SPICE
InceptionV3+ Scene Graph+ Transformer	12.72	1.23	0.0	0.0	3.79	11.47	0.69	0.43
Full Pipeline	15.87	0.54	0.0	0.0	3.92	17.49	0.04	0.00

Table 4.4: Scores using Scene Graph Generator [B-n refers to BLEU-n]

4.2.3 Experiment on Attention on Attention for Image Captioning

In order to compare and analysis, we experimented on one of the pioneer papers of VizWiz-Captions [5] challenge. The generated test outputs are uploaded to the evaluation server for generating the performance scores. The environmental setup is shown in table 4.5.

Dataset	VizWiz-Captions
Batch Size	64
train epoch	25
Data points	155,000
Beam Search	3
Vocab Size	7279
Learning Rate	0.0002
Tested	Evaluation Server
System	Google Colab

Table 4.5: Environmental Setup for AoA-Net [9]

Results & Findings

We tested this model after 7 epochs and 25 epochs of training on the evaluation server hosted in https://eval.ai. The scores are shown in figure 4.6. The resulted scores are nearly same as the author's.

Model	B-1	B-2	B-3	B-4	METEOR	ROUGE-L	CIDEr	SPICE
Implemented (7 epoch)	63.59	45.05	31.44	21.69	18.19	43.91	47.96	13.26
Implemented (25 epoch)	63.9	45.69	32.16	22.41	18.76	44.63	53.77	14.12
Authors	65.91	47.77	33.68	23.41	20.00	46.58	59.77	15.11

Table 4.6: Comparative view on Regenerated scores with authors score [B-n refers to BLEU-n]

Findings

- Because of RNN, it took more time during training and Inference.
- Generated captions were missing plain texts in the images.
- Found miss-classification of objects in the generated captions
- Dependent on Pre-trained bottom up feature extractor.

4.2.4 Experiment on One For All (OFA) Model

So far our experiments were done on models focused on visually challenged image datasets. To see how a state of the art model of fine grained image captioning results on visually challenged image captioning we run inference on OFA model [10] using their cleaned caption generation weight. We only use the test split of VizWiz on OFA to generate captions and tested them on the eval.ai automated testing server. The environmental setup is shown in table 4.7.

Dataset	VizWiz-Captions
Batch Size	1
Data point	8000 (test split)
Tested	Evaluation Server
System	Google Colab

Table 4.7: Environmental setup for experiment on One for all [10] model

Results

Corresponding scores are shown in table 4.8. We can see that it performed similarly with AoA-Net. On MSCOCO caption this model beat every other model but here it could not beat the LSTM based AoA-Net. It states that for captioning visually challenged images models need extra care.

Model	B-1	B-2	B-3	B-4	METEOR	ROUGE-L	CIDEr	SPICE
Pretrained OFA	63.07	44.28	30.29	20.30	18.23	43.65	54.94	12.96

Table 4.8: Score generated by pretrained OFA model [B-n refers to BLEU-n]

4.3 Result Analysis

The overall results and comparison with other models are shown in table 4.9. We had 4 combinations of our proposed pipe-line. The best accuracy was found using only Global feature extractor and OCR token module. All though our best results are not up to the mark, our approach shows the importance of OCR tokens. Without any kind of model ensembling the best single model performance is found with AoA-Net [9].

Our proposed pipeline didn't perform well though it was supposed to do well. The main reason of this failure is the inconsistent Scene graph generator. Since we used pre-trained scene graph we didn't have any control on it. A more detailed explanation is given in discussion session. Our best accuracy is Bleu-4 6.42, METEOR 11.75, ROUGE-L 29.11, CIDEr 13.68 and SPICE 5.86. which very less compared to the AoA-Net result having Bleu-4 23.41, METEOR 20.00, ROUGE-L 46.58, CIDEr 59.77 and SPICE 15.11. We have also experimented general purpose captioning model OFA [10] which is currently state of the art on MSCOCO [11]. We can see this sota model also suffers and couldn't beat the AoA on VizWiz-Caption dataset.

4.4 Discussions

Use of OCR token which were generated from the plain text in the image(if any) has a great impact on caption generation. Our experiment shows that if we incorporate OCR token the test scores increase a lot. Beside that detecting object correctly is the building block of caption generation. Our proposed pipeline was expected to generate better result with the use of scene

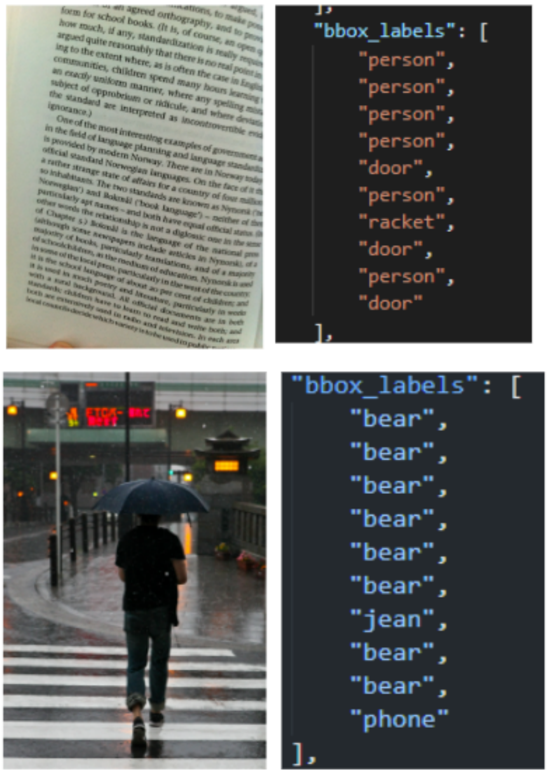
Model	**B-1**	**B-2**	**B-3**	**B-4**	**METEOR**	**ROUGE-L**	**CIDEr**	**SPICE**
InceptionV3+ Transformer	31.04	16.51	8.57	4.43	8.27	20.97	14.44	5.69
InceptionV3+ OCR+ Transformer	**40.06**	**22.06**	**12.22**	**6.42**	**11.75**	**29.11**	**13.68**	**5.86**
InceptionV3+ Scene Graph+ Transformer	12.72	1.23	0.0	0.0	3.79	11.47	0.69	0.43
Full Pipeline	15.87	0.54	0.0	0.0	3.92	17.49	0.04	0.00
Pretrained OFA	63.07	44.28	30.29	20.30	18.23	43.65	54.94	12.96
AoA-Net	**65.91**	**47.77**	**33.68**	**23.41**	**20.00**	**46.58**	**59.77**	**15.11**

Table 4.9: Overall result analysis

graph generator. But it failed because the pre-trained Scene graph generator couldn't detect the objects correctly. Two examples are shown in figure 4.1 where we can see in one image there is a open book but scene graph generator detected a lot of persons in there. In other image a man is holding an umbrella but generator detected bear and jeans.

Since scene graph generator failed to detect object correctly relations are also wrong which resulted in wrong caption generation. Possible reasons of wrong object detections are:

- Image quality and content differs very much from visual gnome.
- Motion Blur, unfocussed objects, lack of luminance, strong flashlights, partially visible object may be the cause of mismatched object detection.



```
],
"bbox_labels": [
    "person",
    "person",
    "person",
    "person",
    "door",
    "person",
    "racket",
    "door",
    "person",
    "door"
],
```

```
"bbox_labels": [
    "bear",
    "bear",
    "bear",
    "bear",
    "bear",
    "bear",
    "jean",
    "bear",
    "bear",
    "phone"
],
```

Input Image Detected Object

Figure 4.1: Sample Object detection by Scene Graph Generator

Chapter 5

Conclusion and Future Work

Image captioning is fascinating field that can contribute in several fields from scene understanding to navigation. Even though a lot of work has been done in this field before, the accuracy can still be improved. In our work we tried to find out some research opportunities and propose a system to overcome those to increase the accuracy of image captioning systems.

Image captioning mainly comprises of image processing and language processing. We researched on the image processing techniques that are used for necessary tasks such as feature extraction, object detection etc. We also researched on the language models that are used to generate the final caption. We had to go through a lot of works before finding the research opportunities. After a lot of brainstorming and research we came up with a pipeline that we believed would solve the research opportunities we discovered. We implemented and did some tuning on our model to see how changes in the pipeline effect the result. We were restricted by time and hardware resources but within our limit we explored a lot of opportunities.

Our final proposed pipeline consists of existing state of the art models of scene graph generation, global feature extractor and OCR token extraction, expecting better image captioning. Our experiment have shown that plain text in image is crucial for image description generation. Whenever we incorporated OCR tokens resulting score increased significantly. But unfortunately one of our core module, scene graph generator, failed to detect object correctly which caused a large scale decrease in our results in all the evaluation metrics. Though our pipeline couldn't beat state of the art works, it opened some new opportunities to find what could have gone wrong and possible research gaps and how to overcome them. Beside that we also showed that general purpose image captioning system cannot work well for visually challenged image captioning as the images captured by visually impaired people differs by a lot from general images. A lots of work is needed to improve this kind of captioning system.

In our future work we will explore more to find what could have made the pipeline a failure.

Beside that we will try to use some image pre-processing tools such as: DeblurGan for removing motion blur, Image sharpening tools, Shadow removing and luminance correction tools. As most of the images used used in our experiments were not well focused and in some of the cases blurry, we believe these image pre-processing tools will improve our result. We also plan to reduce the complexity of our model with reasonable accuracy compromising for small scale devices.

Finally, even though we couldn't improve on the state of the art results, we believe we made a significant contribution in this field through our research and opened a lot of opportunities for future work. With enough resources available in future, we plan to continue our work and increase the accuracy of our system in the coming days.

Bibliography

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.

[2] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 684–699, 2018.

[3] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7008–7024, 2017.

[4] T. Yao, Y. Pan, Y. Li, Z. Qiu, and T. Mei, "Boosting image captioning with attributes," in *Proceedings of the IEEE international conference on computer vision*, pp. 4894–4902, 2017.

[5] D. Gurari, Y. Zhao, M. Zhang, and N. Bhattacharya, "Captioning images taken by people who are blind," in *European Conference on Computer Vision*, pp. 417–434, Springer, 2020.

[6] P. Dognin, I. Melnyk, Y. Mroueh, I. Padhi, M. Rigotti, J. Ross, Y. Schiff, R. A. Young, and B. Belgodere, "Image captioning as an assistive technology: Lessons learned from vizwiz 2020 challenge," *Journal of Artificial Intelligence Research*, vol. 73, pp. 437–459, 2022.

[7] D. W. Kim, J. gwon Hwang, S. H. Lim, and S. H. Lee, "An improved feature extraction approach to image captioning for visually impaired people,"

[8] X. Yang, Y. Liu, and X. Wang, "Reformer: The relational transformer for image captioning," *arXiv preprint arXiv:2107.14178*, 2021.

[9] L. Huang, W. Wang, J. Chen, and X.-Y. Wei, "Attention on attention for image captioning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4634–4643, 2019.

[10] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," *arXiv preprint arXiv:2202.03052*, 2022.

[11] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv preprint arXiv:1504.00325*, 2015.

[12] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Computing Surveys (CsUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[13] G. Letarte, F. Paradis, P. Giguère, and F. Laviolette, "Importance of self-attention for sentiment analysis," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 267–275, 2018.

[14] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3716–3725, 2020.

[15] P. Ackland, S. Resnikoff, and R. Bourne, "World blindness and visual impairment: despite many successes, the problem is growing," *Community Eye Health*, vol. 30, pp. 71–73, 01 2017.

[16] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, and R. Cucchiara, "From show to tell: A survey on image captioning," 07 2021.

[17] Q. Zhu and J. Luo, "Generative pre-trained transformer for design concept generation: an exploration," *arXiv preprint arXiv:2111.08489*, 2021.

[18] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[19] X. Yang, K. Tang, H. Zhang, and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10685–10694, 2019.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[22] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, no. 1, pp. 32–73, 2017.

[23] K. Erk, "Vector space models of word meaning and phrase meaning: A survey," *Language and Linguistics Compass*, vol. 6, no. 10, pp. 635–653, 2012.

[24] Y. Goldberg and O. Levy, "word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, 2014.

[25] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[28] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[31] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," *arXiv preprint arXiv:1906.04341*, 2019.

[32] "What are the types of RNN?." `https://www.educative.io/edpresso/what-are-the-types-of-rnn`. Accessed: 2022-4-14.

[33] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[36] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015.

[37] T. Yao, Y. Pan, Y. Li, and T. Mei, "Hierarchy parsing for image captioning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2621–2629, 2019.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[40] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[41] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

[42] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

[43] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, *et al.*, "From captions to visual concepts and back," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1473–1482, 2015.

[44] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015.

[45] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, pp. 2048–2057, PMLR, 2015.

[46] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo, "Attentive relational networks for mapping images to scene graphs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3957–3966, 2019.

[47] "PaddleOCR: Awesome multilingual OCR toolkits based on PaddlePaddle (practical ultra lightweight OCR system, support 80+ languages recognition, provide data annotation and synthesis tools, support training and deployment among server, mobile, embedded and IoT devices)."

[48] M. Rahman, N. Mohammed, N. Mansoor, and S. Momen, "Chittron: An automatic bangla image captioning system," *Procedia Computer Science*, vol. 154, pp. 636–642, 2019.

[49] O. Pelka, S. Koitka, J. Rückert, F. Nensa, and C. M. Friedrich, "Radiology objects in context (roco): a multimodal image dataset," in *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, pp. 180–189, Springer, 2018.

[50] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, pp. 74–81, 2004.

[51] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *European conference on computer vision*, pp. 382–398, Springer, 2016.